

## NAVIGASI BERBASIS *BEHAVIOR* DAN *FUZZY LOGIC* PADA SIMULASI ROBOT BERGERAK OTONOM

**Rendyansyah<sup>\*</sup>, Kemahyanto Exaudi, Aditya Putra Perdana Prasetyo**

Jurusan Sistem Komputer, Fakultas Ilmu Komputer, Universitas Sriwijaya

<sup>\*</sup>Corresponding author, e-mail : rendyansyah@ilkom.unsri.ac.id

**Abstrak**— *Mobile robot* merupakan mekanisme robot yang mampu bergerak otomatis. Pergerakan robot secara otomatis memerlukan suatu sistem navigasi. Navigasi adalah metode untuk menentukan gerak robot. Pada penelitian ini navigasi robot dikembangkan menggunakan metode *behavior* (perilaku) dengan logika *fuzzy*. Perilaku robot dibagi menjadi beberapa modul, seperti berjalan, menghindari halangan, mengikuti dinding, koridor maupun kondisi *u-shape*. Pada penelitian ini dirancang simulasi *mobile robot* di dalam pemrograman visual. Robot dilengkapi dengan tujuh sensor jarak dan dibagi menjadi beberapa kelompok untuk menguji perilaku yang dirancang, sehingga perilaku robot menghasilkan pengaturan kecepatan dan *steering*. Berdasarkan percobaan yang telah dilakukan menunjukkan bahwa simulasi *mobile robot* dapat berjalan mulus (*smooth*) pada berbagai kondisi. Hal ini membuktikan bahwa implementasi pembentukan *behavior* dan teknik logika *fuzzy* pada robot bekerja dengan baik.

*Kata Kunci* : *behavior, logika fuzzy, mobile robot*

**Abstract**— *Mobile robot* is the robotic mechanism that is able to moved automatically. The movement of the robot automatically require a navigation system. Navigation is a method for determining the robot motion. In this study, using a method developed robot navigation behavior with fuzzy logic. The behavior of the robot is divided into several modules, such as walking, avoid obstacles, to follow walls, corridors and conditions of u-shape. In this research designed mobile robot simulation in a visual programming. Robot is equipped with seven distance sensor and divided into several groups to test the behavior that is designed, so that the behavior of the robot generate speed and steering control. Based on experiments that have been conducted shows that mobile robot simulation can run smooth on many conditions. This proves that the implementation of the formation of behavior and fuzzy logic techniques on the robot working well.

*Keywords* : *behavior, fuzzy logic, mobile robot*

*Copyright © 2016 JNTE. All rights reserved*

### 1. PENDAHULUAN

*Mobile robot* dirancang untuk bergerak secara otomatis atau dikenal robot bergerak otonom, robot yang dapat bergerak bebas pada area yang tidak terstruktur dan dapat menentukan sendiri jalur pergerakannya. Robot otonom identik sebagai robot cerdas karena sistem-nya menggunakan teknik kendali cerdas (*artificial intelligence*) untuk mengontrol pergerakan. Kontrol gerakan robot ini dikenal istilah navigasi cerdas [1].

Navigasi merupakan suatu metode dalam menentukan aksi dari pergerakan robot di dalam lingkungan [1]. Navigasi cerdas diaplikasikan pada robot bergerak otonom agar dapat membuat keputusan sendiri. Saat ini banyak dikembangkan suatu sistem navigasi robot bergerak otonom menggunakan logika *fuzzy*

[2][3][4], teknik jaringan saraf [1][4], bahkan gabungan dari keduanya (*neuro-fuzzy*) [5]. Metode yang paling sering digunakan yaitu logika *fuzzy* karena proses *fuzzy* bisa diimplementasikan ke dalam sistem mikro [2][6]. Algoritma logika *fuzzy* digunakan dalam aplikasi kendali yang memerlukan waktu komputasi cepat, oleh karena itu dapat diaplikasikan ke robot dengan banyak *behavior*.

Sistem navigasi cerdas pada robot bergerak otonom dengan menirukan perilaku atau reaksi terhadap keadaan lingkungan yang dihadapi. Metode *behavior* dapat bekerja pada lingkungan yang dinamis, salah satu contoh *behavior* pada robot dalam mencari target [2][7]. Metode *behavior* terinspirasi dari sifat perilaku makhluk hidup saat berinteraksi dengan lingkungannya [8]. *Behavior* tersebut terjadi karena alat indra menerima informasi sehingga ada respon gerak.

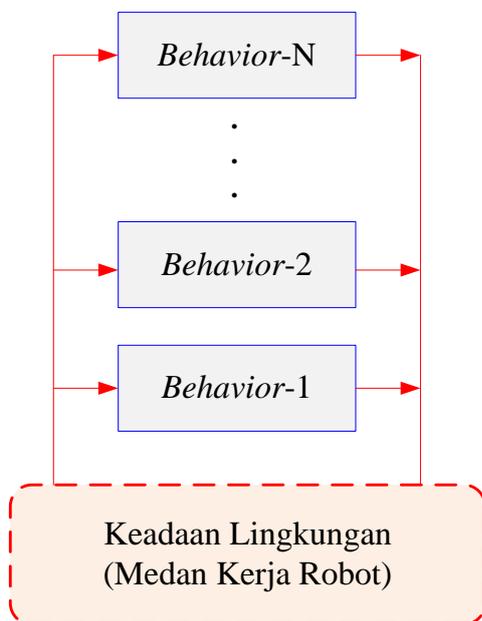
Dalam kasus robot bergerak otonom, informasi diperoleh dari sensor dan respon keluaran berupa pergerakan.

Pada penelitian ini diimplementasikan *behavior* sebagai penentu aksi gerak, dan *fuzzy logic* sebagai kendali cerdas untuk mengontrol posisi dan kecepatan robot. Kombinasi tersebut bertujuan untuk menghasilkan respon pergerakan yang halus (*smooth*) dalam bernavigasi pada lingkungan komplek yang tak terstruktur. Pergerakan *mobile robot* dan lingkungan disimulasikan ke dalam tampilan visual.

## 2. TINJAUAN PUSTAKA

### 2.1. Metode Berbasis Behavior

Metode *behavior* juga dikenal sebagai arsitektur *subsumption* merupakan perilaku yang meniru sifat alami makhluk hidup dan tergolong ke dalam algoritma *high-level control* [8]. Adapun ilustrasi dari arsitektur *behavior* ditunjukkan pada Gambar 1.



Gambar 1. Arsitektur *behavior* [8].

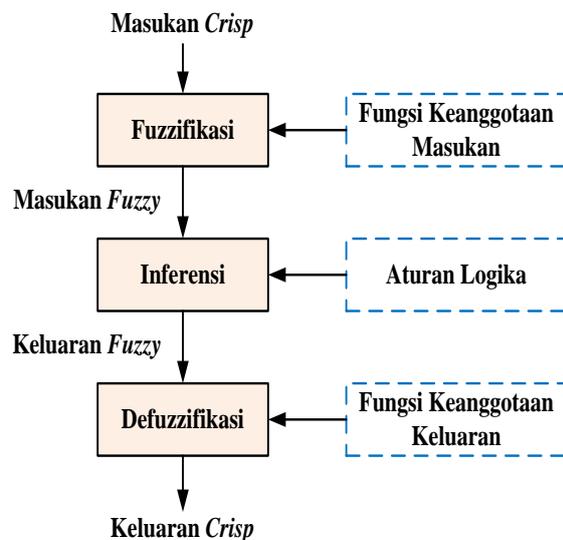
Berdasarkan arsitektur pada Gambar 1, *behavior* dapat diterapkan pada robot bergerak otonom dengan membentuk tingkat *behavior*. Adapun tingkat *behavior* menentukan prioritas, semakin tinggi tingkat *behavior* maka dapat men-skip tingkat yang lebih rendah. Tingkat *behavior* tersebut terjadi karena adanya stimuli

atau rangsangan yang didapat dari lingkungan. *Behavior* pada robot dirancang berdasarkan strategi-strategi yang diinginkan dalam menemukan targetnya [9].

Berdasarkan strategi *behavior* ini, maka penulis mengaplikasikan teknik *behavior* pada robot untuk membagi tugas robot dalam bernavigasi.

### 2.2. Algoritma Logika Fuzzy

Logika *fuzzy* bekerja atas dasar logika dan keinginan dari seorang ahli [6][10]. Pada perancangan logika *fuzzy* memiliki beberapa tahapan [11][12], yaitu : (1) fuzzifikasi, (2) basis aturan dan mekanisme inference, dan (3) defuzzifikasi. Gambar 2 memperlihatkan diagram blok algoritma logika *fuzzy*.



Gambar 2. Diagram blok logika *fuzzy* [13].

Penjelasan tahapan algoritma logika *fuzzy* :

1. Fuzzifikasi.  
Fuzzifikasi akan mengubah nilai masukan *crisp* menjadi nilai *fuzzy*. Nilai *fuzzy* dipengaruhi oleh fungsi keanggotaan yang dirancang oleh pakar.
2. Basis aturan dan mekanisme inference.  
Aturan (*rule*) berisi logika-logika dari seorang pakar dalam menentukan aksi sistem. Oleh karena itu basis aturan yang ditetapkan dengan mengintegrasikan variabel masukan dengan variabel keluaran fungsi keanggotaan. Aturan ini berupa logika “jika – maka” seperti :  
*if X<sub>1</sub> is A<sub>1</sub> and ... and X<sub>n</sub> is A<sub>n</sub>, then Y is B*

Aturan-aturan tersebut dibutuhkan penalaran dalam pengambilan keputusan akhir, sehingga pada tahap ini akan menghasilkan keluaran *fuzzy*. Mekanisme dalam pengambilan keputusan menggunakan metode “Max-Min” seperti :

$$\mu_B(y) = \max \left[ \min \left[ \mu_{A1}(input(i)), \mu_{A2}(input(j)), \dots \right] \right]$$

3. Defuzzifikasi.

Defuzzifikasi akan mengkonversi nilai keluaran *fuzzy* menjadi nilai keluaran *crisp*. Proses ini juga dirancang fungsi keanggotaan keluaran. Metode umum yang sering dipakai dalam komputasi defuzzifikasi yaitu metode *centroid* atau *weighted average* [12][14].

3. METODOLOGI

Sistem robotik dalam penelitian ini menggunakan simulasi di dalam program visual. Oleh karena itu, dibuatlah rancangan navigasi berbasis *behavior* dan logika *fuzzy* pada simulasi tersebut.

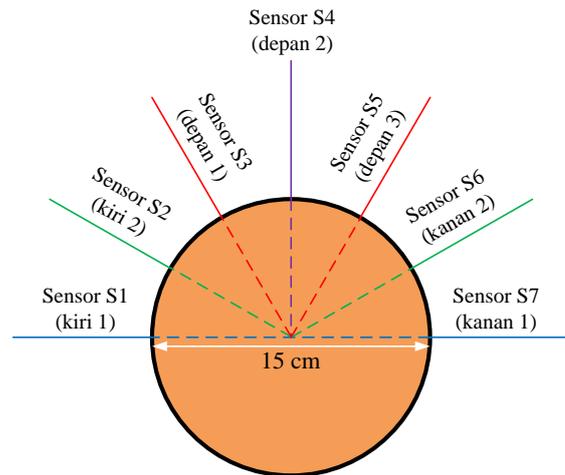
3.1. Perancangan *Behavior*

Di dalam simulasi, robot berbentuk bulat dengan diameter 15 cm dan dilengkapi tujuh sensor jarak, penggerak kecepatan dan sudut belok (*steering*). Gambar 3 menunjukkan bentuk robot dan peletakan sensor jarak. Simulasi robot dijalankan dalam arena pada Gambar 4. *Behavior* pada robot dalam menavigasi dikelompokkan menjadi beberapa kelompok, yaitu: perilaku maju, mengikuti dinding kiri atau kanan, mengikuti jalur koridor, menghindari halangan dan kondisi *u-shape*. Adapun diagram blok *behavior* untuk navigasi robot ditunjukkan pada Gambar 5.

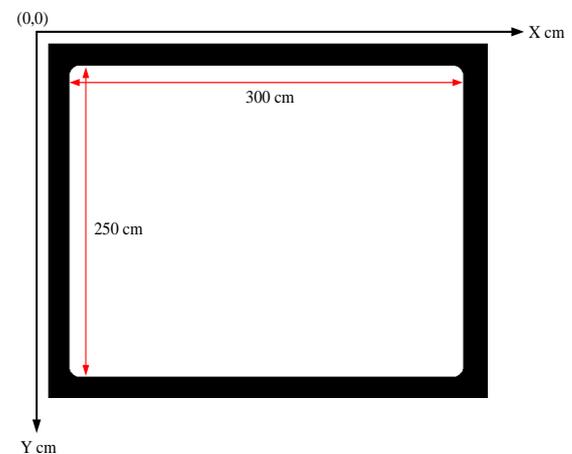
Berdasarkan Gambar 5, bahwa prioritas paling tinggi yaitu *behavior* dalam kondisi terlalu dekat dengan halangan (*u-shape*), dan yang paling rendah pada *behavior* bergerak maju. Adapun *behavior* dalam simulasi robot didefinisikan sebagai berikut :

1. Bergerak maju: robot bergerak bebas kearah depan dengan kecepatan konstan, karena tidak ada halangan atau halangan dianggap jauh.
2. Mengikuti dinding kanan: ikuti jika terdeteksi dinding kanan. Jika jarak yang dideteksi oleh sensor S<sub>6</sub> dan S<sub>7</sub> masing-

masing kurang dari 40 cm dan 30 cm maka robot akan mengikuti dinding kanan.

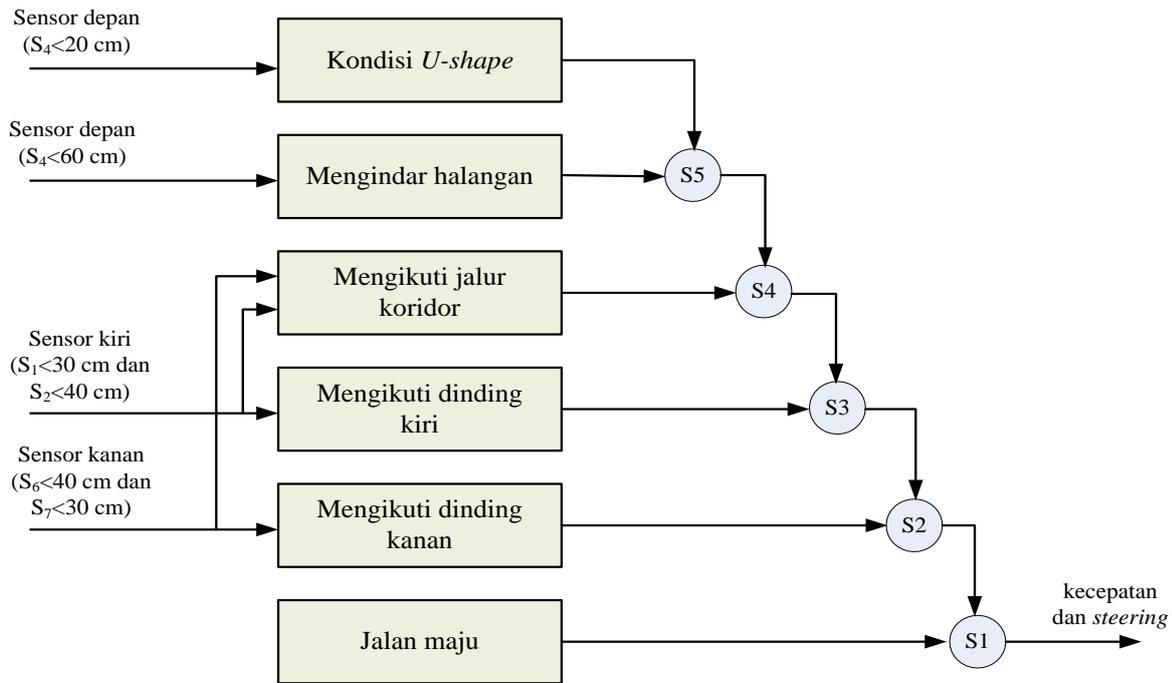


Gambar 3. Bentuk robot dan peletakan tujuh sensor jarak.



Gambar 4. Simulasi arena robot.

3. Mengikuti dinding kiri: ikuti jika terdeteksi dinding kiri. Jika jarak yang dideteksi oleh sensor S<sub>1</sub> dan S<sub>2</sub> masing-masing kurang dari 30 cm dan 40 cm maka robot akan mengikuti dinding kiri.
4. Mengikuti jalur koridor: jika sensor kanan (S<sub>6</sub>) dan kiri (S<sub>2</sub>) mendeteksi jarak kurang dari 40 cm, artinya robot menganggap jalur koridor. Robot akan berjalan pada posisi tengah jalur koridor tersebut.
5. Menghindar halangan: Hindari jika ada halangan depan. Prilaku ini diaktifkan saat sensor depan (S<sub>4</sub>) mendeteksi jarak kurang dari 60 cm. Gerakan belok kiri atau kanan berdasarkan informasi dari sensor S<sub>3</sub> dan S<sub>5</sub>.



Gambar 5. Arsitektur *behavior* untuk navigasi pada robot dalam simulasi.

6. Kondisi *U-shape*: Jika hanya sensor  $S_4$  mendeteksi jarak yang terlalu dekat (kurang dari 20 cm) maka robot ber-manuver kearah kiri ataupun kanan berdasarkan informasi dari sensor  $S_1$  dan  $S_7$ . Jika sensor  $S_1 < S_7$  maka belok kanan, dan sebaliknya.

Pada *behavior* dalam navigasi robot, penulis menambahkan logika *fuzzy* guna menghasilkan gerak robot yang baik. Logika *fuzzy* diimplementasikan pada beberapa *behavior* diantaranya menghindari halangan, mengikuti dinding kanan atau kiri, dan mengikuti koridor.

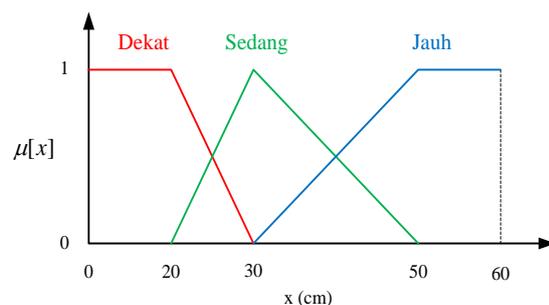
**3.2. Perancangan Logika Fuzzy**

Logika *fuzzy* pada *behavior* robot bekerja berdasarkan informasi sensor-sensor jarak untuk mengendalikan gerak kecepatan dan *steering*. Adapun rancangan logika *fuzzy* untuk *behavior* robot :

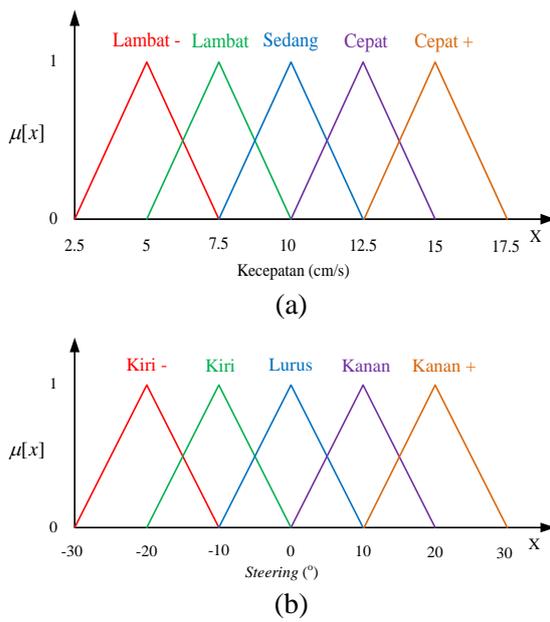
**3.2.1. Behavior menghindari halangan**

Masukan *crisp* dari *behavior* ini yaitu tiga sensor jarak pada bagian depan robot (kiri depan ( $S_3$ ), depan ( $S_4$ ), kanan depan ( $S_5$ )). Keluaran dari logika *fuzzy* pada *behavior* ini menghasilkan *crisp* respon gerak kecepatan dan *steering*. Proses rancangan fungsi keanggotaan untuk tiga sensor jarak ini dengan melakukan tahap *trial*

*and error* untuk mengetahui respon robot terhadap halangan. Pada *behavior* ini tiga sensor jarak tersebut dibatasi jarak maksimum 60 cm, sehingga diperoleh masukan jarak yang dibagi menjadi tiga variabel linguistik yaitu dekat, sedang dan jauh. Jika terdeteksi jarak dalam area dekat maka robot harus ber-manuver besar kearah tempat yang tidak ada halangan. Jika jarak sedang maka robot ber-manuver sedang. Jika jaraknya jauh maka robot ber-manuver sedikit. Bentuk dan ukuran robot juga mempengaruhi respon gerak, misal robot berukuran lebih besar atau kecil maka jarak maksimumnya bisa diganti. Adapun fungsi keanggotaan untuk masukan nilai jarak ditunjukkan pada Gambar 6.



Gambar 6. Fungsi keanggotaan untuk masukan jarak pada sensor  $S_3$ ,  $S_4$  dan  $S_5$ .



Gambar 7. Fungsi keanggotaan untuk keluaran. (a) kecepatan, (b) steering.

Keluaran respon kecepatan dan steering masing-masing terdiri lima variabel linguistik seperti pada Gambar 7. Berdasarkan Gambar 7 bahwa rentang kecepatan yang diberikan pada robot yaitu 2.5 cm/s sampai 17.5 cm/s dan steering -30° sampai 30°. Pengambilan keputusan pada fungsi keanggotaan keluaran menggunakan *scaled fuzzified decision*, dan digunakan juga pada *behavior* lain.

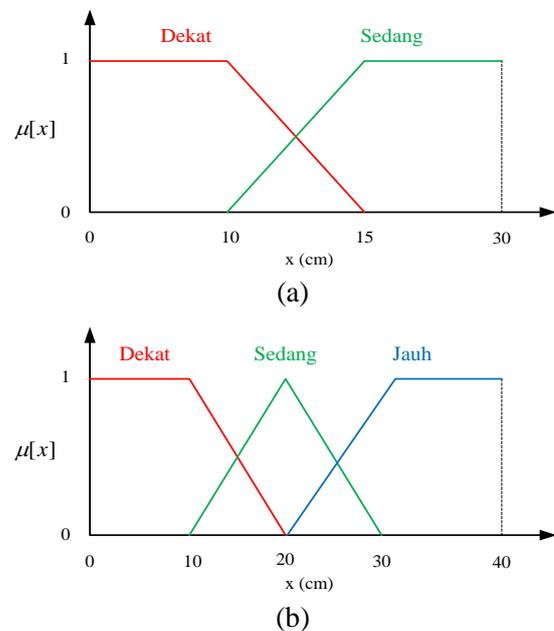
Tabel 1. Aturan logika untuk menghindari halangan.

		Sensor jarak (S <sub>3</sub> )		
		Dekat	Sedang	Jauh
Sensor Jarak (S <sub>4</sub> dan S <sub>5</sub> )	Dekat, Dekat	Lambat-	Lambat-	Lambat
	Dekat, Dekat	Lurus	Kiri-	Kiri
	Dekat, Sedang	Lambat-	Lambat-	Lambat-
	Dekat, Sedang	Kanan+	Lurus	Kiri
	Dekat, Jauh	Lambat	Lambat-	Lambat-
	Dekat, Jauh	Kanan	Kanan	Lurus
	Sedang, Dekat	Lambat-	Lambat	Sedang
	Sedang, Dekat	Lurus	Kiri	Kiri
	Sedang, Sedang	Lambat	Sedang	Cepat
	Sedang, Sedang	Kanan	Lurus	Kiri
	Sedang, Jauh	Sedang	Cepat	Cepat
	Sedang, Jauh	Kanan	Kanan	Lurus
Jauh, Dekat	Lambat	Sedang	Cepat	
Jauh, Dekat	Lurus	Kiri	Kiri	
Jauh, Sedang	Sedang	Cepat	Cepat	
Jauh, Sedang	Kanan	Lurus	Lurus	
Jauh, Jauh	Cepat	Cepat	Cepat+	
Jauh, Jauh	Kanan	Lurus	Lurus	

Berdasarkan kombinasi dari informasi fungsi keanggotaan masukan dan keluaran, menghasilkan 27 aturan logika untuk pergerakan robot menghindari halangan. Adapun aturan logika untuk menghindari halangan seperti ditunjukkan pada Tabel 1.

3.2.2. Behavior mengikuti dinding kanan atau kiri

Behavior ini diaktifkan saat sensor jarak bagian kanan robot mendeteksi dinding kanan, begitu juga untuk sensor bagian kiri. Masukan dari sensor kanan yaitu S<sub>7</sub> dan S<sub>6</sub>, atau kiri yaitu S<sub>1</sub> dan S<sub>2</sub>. Proses rancangan fungsi keanggotaan sensor jarak juga dilakukan tahap *trial and error*. Dalam kasus ini, sensor S<sub>1</sub> dan S<sub>7</sub> terdiri atas variabel dekat dan sedang dengan jarak maksimum 30 cm, untuk sensor S<sub>2</sub> dan S<sub>6</sub> dengan variabel dekat, sedang dan jauh dengan jarak maksimum 40 cm. Jika jarak dalam area dekat maka robot harus belok menjauhi dinding, dan jika jarak dalam area jauh maka robot harus belok mendekati dinding. Adapun fungsi keanggotaan untuk nilai jarak ditunjukkan pada Gambar 8. Behavior ini juga menghasilkan keluaran *crisp* untuk kecepatan dan steering. Untuk keluaran kecepatan dan steering sama seperti pada behavior menghindari halangan, sehingga grafik fungsi keanggotaannya juga digunakan sama seperti pada Gambar 7.



Gambar 8. Fungsi keanggotaan untuk masukan jarak, (a) sensor S<sub>1</sub> dan S<sub>7</sub>, (b) sensor S<sub>2</sub> dan S<sub>6</sub>.

Kombinasi dari masukan dan keluaran pada *behavior* mengikuti dinding kanan ataupun kiri masing-masing menghasilkan enam aturan logika. Adapun aturan logika mengikuti dinding kanan seperti pada Tabel 2, dan aturan logika mengikuti dinding kiri pada Tabel 3.

Tabel 2. Aturan logika untuk mengikuti dinding kanan.

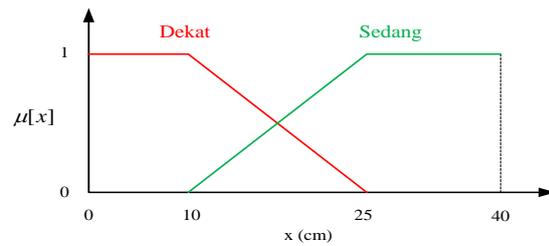
		Sensor jarak (S <sub>7</sub> )	
		Dekat	Sedang
Sensor Jarak (S <sub>6</sub> )	Dekat	Lambat- Kiri-	Sedang Kiri
		Sedang	Lambat Kanan
	Jauh		Lambat Kanan

Tabel 3. Aturan logika untuk mengikuti dinding kiri.

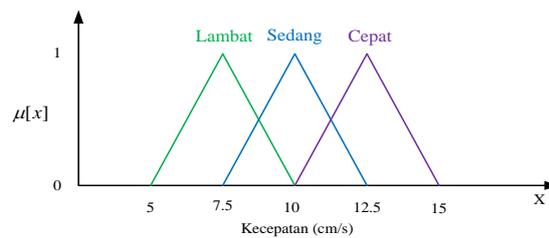
		Sensor jarak (S <sub>1</sub> )	
		Dekat	Sedang
Sensor Jarak (S <sub>2</sub> )	Dekat	Lambat- Kanan+	Sedang Kanan
		Sedang	Lambat Kiri
	Jauh		Lambat Kiri

**3.2.3. Behavior mengikuti jalur koridor**

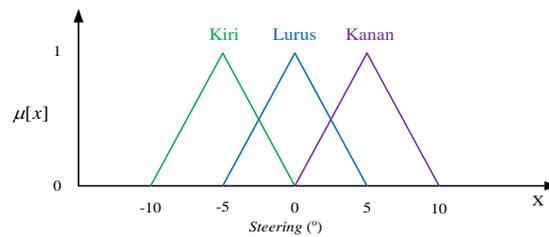
Mengikuti koridor berarti robot berada diantara dinding kanan dan kiri. Pada kondisi ini robot akan bergerak pada posisi tengah jalur koridor. Dengan kata lain sensor jarak S<sub>2</sub> dan S<sub>6</sub> sama-sama aktif mendeteksi dinding dengan batas maksimum 40 cm, dan juga mengontrol pergerakan robot. Pada *behavior* ini rancangan fungsi keanggotaan juga berdasarkan *trial and error*. Jika informasi jarak pada sensor S<sub>2</sub> dekat maka sensor S<sub>6</sub> sedang, dan sebaliknya. Ketika terdeteksi jarak oleh sensor S<sub>2</sub> dekat dan sensor S<sub>6</sub> sedang maka robot manuver kekanan, namun jika sensor S<sub>2</sub> sedang dan sensor S<sub>6</sub> dekat maka robot manuver kekiri. Oleh karena itu diputuskan variabel linguistiknya dekat dan sedang. Gambar 9 memperlihatkan fungsi keanggotaan masukan nilai jarak untuk sensor S<sub>2</sub> dan S<sub>6</sub>. Adapun fungsi keanggotaan keluaran untuk kecepatan dan *steering* ditunjukkan pada gambar 10. Berbeda dari *behavior* sebelumnya, pada *behavior* ini rentang kecepatan 5 cm/s sampai 15 cm/s dan *steering* -10° sampai 10°.



Gambar 9. Fungsi keanggotaan untuk masukan nilai jarak pada sensor S<sub>2</sub> dan S<sub>6</sub>.



(a)



(b)

Gambar 10. Fungsi keanggotaan untuk keluaran, (a) kecepatan, dan (b) *steering*.

Kombinasi dari masukan jarak dengan keluaran respon robot menghasilkan aturan logika seperti yang ditunjukkan pada Tabel 4.

Tabel 4. Aturan logika untuk mengikuti jalur koridor

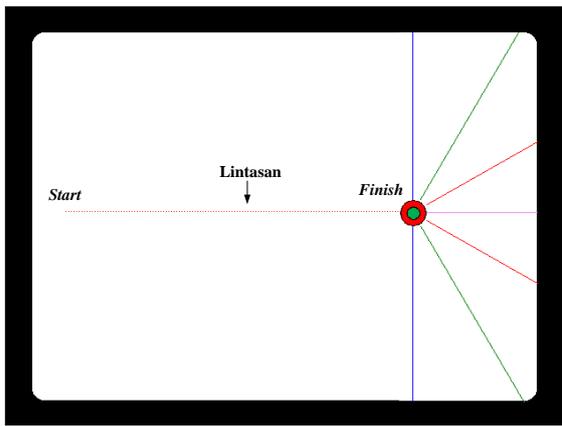
		Sensor jarak (S <sub>2</sub> )	
		Dekat	Sedang
Sensor Jarak (S <sub>6</sub> )	Dekat	Lambat	Sedang
		Lurus	Kiri
	Sedang	Sedang	Cepat
		Kanan	Lurus

**4. HASIL DAN PEMBAHASAN**

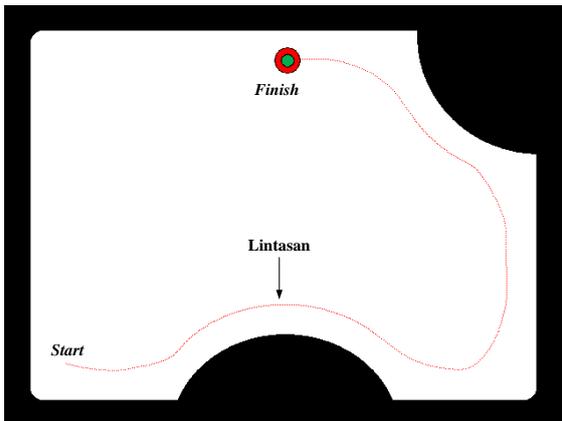
Pengujian simulasi dilakukan dengan menempatkan robot dalam lingkungan yang ditentukan, seperti lingkungan untuk robot mengikuti dinding kanan atau kiri, koridor, menghindari halangan bahkan *u-shape*.

#### 4.1. Pengujian robot pada lingkungan bebas

Pada pengujian ini robot dijalankan pada lingkungan bebas yang tidak ada halangan. Sensor pada robot tidak mendeteksi jarak atau jarak dianggap jauh, sehingga robot bergerak maju berdasarkan *behavior* yang diterapkan dengan kecepatan konstan 15 cm/s. Oleh karena belum ada obyek/halangan maka *behavior* pada robot berjalan maju bersifat prioritas rendah. Ilustrasi pergerakan robot dalam lingkungan bebas seperti ditunjukkan pada Gambar 11. Lintasan merupakan jalur yang telah dilalui oleh robot saat ber-navigasi dalam lingkungan.



Gambar 11. Respon pergerakan robot dalam lingkungan bebas.



Gambar 12. Respon pergerakan robot dalam mengikuti dinding kanan.

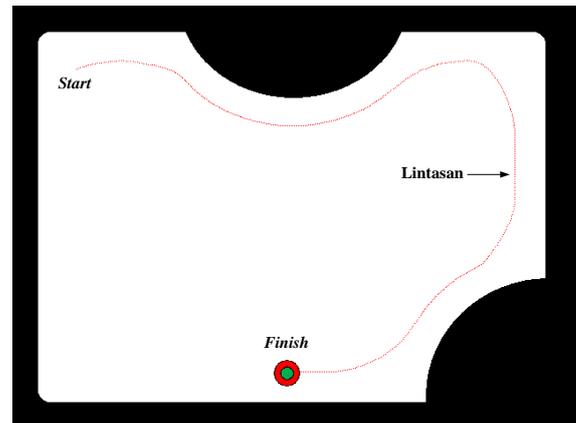
#### 4.2. Pengujian robot dalam mengikuti dinding kanan

Pada pengujian ini bagian kanan robot ditempatkan di dekat dinding, kemudian robot dijalankan untuk melihat respon pergerakannya

dalam mengikuti dinding kanan. *Behavior* ini aktif karena ada stimuli sensor bagian kanan robot mendeteksi dinding kanan. Pergerakan robot ini senantiasa mengikuti dinding dengan mempertahankan jarak terhadap dinding yaitu  $\pm 18$  cm. Hal ini karena logika *fuzzy* bekerja dengan baik dalam mengontrol pergerakan robot (kecepatan dan *steering*). Gambar 12 menunjukkan respon pergerakan robot dalam mengikuti dinding kanan.

#### 4.3. Pengujian robot dalam mengikuti dinding kiri

Sama seperti pada robot mengikuti dinding kanan. Robot ini juga diuji untuk mengikuti dinding kiri. Hal ini perlu supaya diketahui respon gerak robot dalam mengikuti dinding kiri. Pada *behavior* ini juga aktif karena ada stimuli sensor bagian kiri robot mendeteksi dinding. Logika *fuzzy* yang ditanam pada *behavior* ini juga memberikan respon gerak yang baik. Robot juga mempertahankan jarak terhadap dinding  $\pm 18$  cm. Gambar 13 menunjukkan respon pergerakan robot dalam mengikuti dinding kiri.

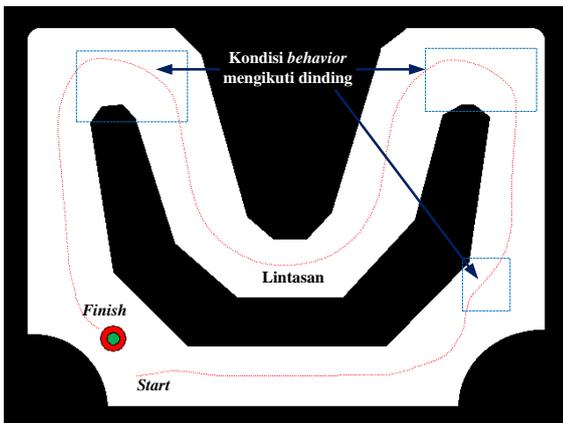


Gambar 13. Respon pergerakan robot dalam mengikuti dinding kiri.

#### 4.4. Pengujian robot pada jalur koridor

Jalur koridor berarti jalur yang diapit oleh dua dinding dengan jarak cukup dekat. Dalam kondisi ini robot berperilaku untuk berjalan mengikuti jalur tersebut. Pergerakan robot dalam *behavior* ini juga dikontrol menggunakan logika *fuzzy*, sehingga robot ber-manuver dengan baik. Dalam kasus ini, jika sensor  $S_2$  atau  $S_6$  mendeteksi jarak lebih dari 40 cm maka robot berperilaku mengikuti dinding (bukan koridor). Oleh karena itu kecepatan dan *steering* juga

berlaku aturan *behavior* mengikuti dinding. Gambar 14 memperlihatkan respon pergerakan robot dalam jalur koridor.



Gambar 14. Respon pergerakan robot dalam jalur koridor.

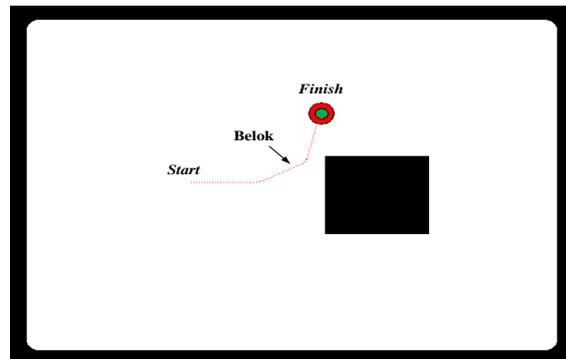
**4.5. Pengujian robot menghindari halangan**

Robot perlu diuji dalam menghindari halangan dengan tujuan untuk mengetahui respon gerakannya. Pengujian robot ini dilakukan dengan obyek berupa kotak, segitiga dan bulat. Obyek tersebut secara bergantian ditempatkan di depan robot. Robot dalam menghindari obyek kotak mulai merespon saat sensor depan mendeteksi jarak 50 cm seperti yang ditunjukkan pada Gambar 15(a). Kasus serupa juga dilakukan pada obyek segitiga, tampak robot juga menghindari halangan seperti terlihat pada Gambar 15(b). Namun berbeda untuk kondisi obyek bentuk bulat, robot juga mendeteksi obyek lalu *behavior* menghindari. Saat robot menghindari (aksi belok kiri) maka sensor bagian kanan ( $S_6$  dan  $S_7$ ) mendeteksi obyek, sehingga *behavior* robot berubah menjadi *right-wall following*. Hal ini juga berlaku jika robot saat belok kanan kemudian sensor kirinya aktif (*left-wall following*). Adapun pergerakan robot dalam menghindari obyek bulat ditunjukkan pada Gambar 15(c).

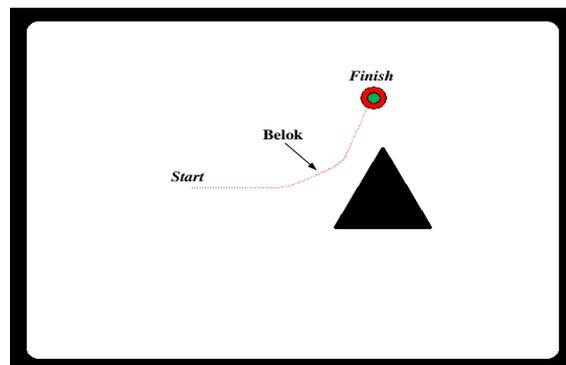
**4.6. Pengujian robot pada lingkungan u-shape**

Adanya *behavior* pada kondisi *u-shape* untuk mengatasi keterbatasan *behavior* menghindari halangan. Pengujian ini juga bertujuan untuk mengetahui respon gerak robot saat dalam kondisi jarak sensor depan ( $S_4$ ) terlalu dekat (kurang dari 20 cm) dengan obyek/dinding (*u-shape*). *Behavior* ini tidak menggunakan

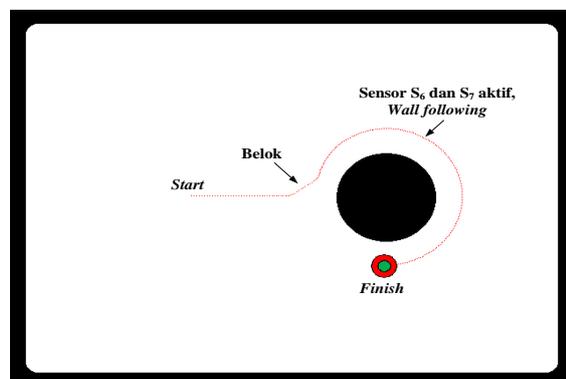
logika *fuzzy* namun logikanya berdasarkan perbandingan dari jarak yang dideteksi oleh sensor  $S_1$  dan  $S_7$ . Pada kondisi ini diberikan kecepatan konstan 5 cm/s dan *steering* konstan  $10^\circ$  (ke kiri atau kanan). Gambar 16 menunjukkan respon pergerakan robot dalam kondisi *u-shape*. Berdasarkan Gambar 16 bahwa robot ber-manuver kearah yang dianggap masih luas. Oleh karena itu, robot selalu ber-navigasi pada lingkungan *u-shape*.



(a)

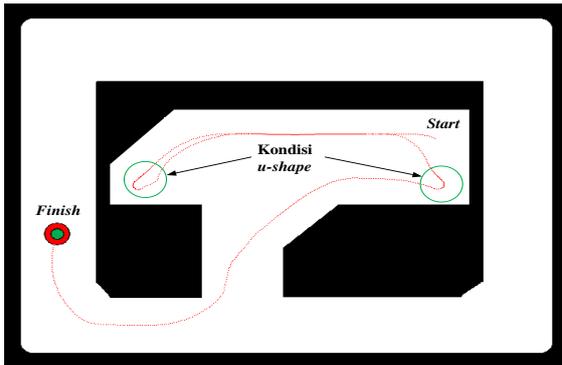


(b)

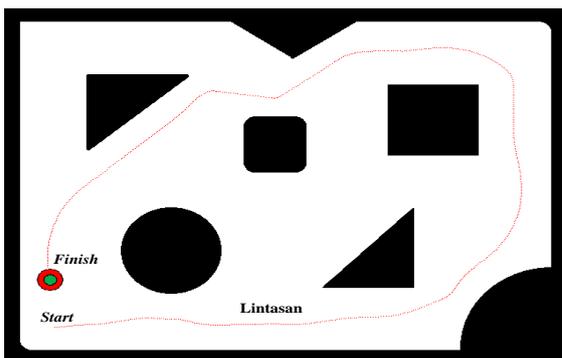


(c)

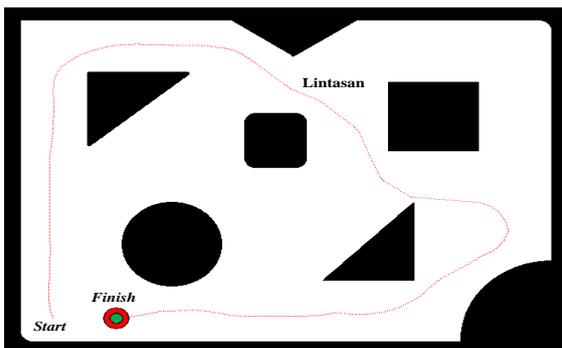
Gambar 15. Respon pergerakan robot dalam menghindari halangan, (a) objek kotak, (b) segitiga, dan (c) bulat.



Gambar 16. Respon pergerakan robot pada kondisi *u-shape*.



(a)



(b)

Gambar 17. Respon gerak robot dalam lingkungan kompleks.  
(a) lintasan robot mengikuti dinding kanan,  
(b) mengikuti dinding kiri.

#### 4.7. Pengujian robot dalam lingkungan kompleks.

Pada lingkungan kompleks simulasi robot ditempatkan dekat dinding (kanan atau kiri). Ini bertujuan agar robot berperilaku mengikuti dinding, namun saat ada halangan maka robot ber-manuver sesuai *behavior* yang diterapkan. Gambar 17 menunjukkan pergerakan robot

(manuver) dalam lingkungan kompleks. Hal ini menunjukkan bahwa integrasi logika *fuzzy* dan *behavior* yang diimplementasikan dalam robot dapat bekerja dengan baik pada lingkungan dengan berbagai kondisi.

## 5. KESIMPULAN

Simulasi *mobile robot* berukuran diameter 15 cm, dan dilengkapi sensor jarak sebanyak tujuh sensor yang diaplikasikan dalam program visual. Sensor-sensor tersebut dibagi menjadi kelompok untuk membentuk beberapa *behavior* dalam manuver robot. Robot telah dilakukan pengujian dalam lingkungan dengan berbagai kondisi. Integrasi logika *fuzzy* dan *behavior* yang diimplementasikan pada robot memberikan hasil yang baik. Pengembangan pada penelitian selanjutnya akan dirancang sistem robot untuk keperluan aplikasi.

## UCAPAN TERIMA KASIH

Terima kasih disampaikan kepada jurusan Sistem Komputer dan Fakultas Ilmu Komputer Universitas Sriwijaya.

## DAFTAR PUSTAKA

- [1] Nurmaini, S. dan Tutuko, B., *A New Classification Technique in Mobile Robot Navigation*, *Telkomnika*, Vol. 9, No. 3, hal 453-464, (2011).
- [2] Sembiring, S., Baafai, U., dan Tarigan, P., *Implementasi Arsitektur Behavior-Based dengan Menggunakan Fuzzy untuk Navigasi Car-Like Mobile Robot dalam Lingkungan yang Tak Dikenal*, *Jurnal Generic*, Vol. 9, No. 2, hal 320-331, (2014).
- [3] Hamzah, M.I. dan Abdall, T.Y., *Mobile Robot Navigation using Fuzzy Logic and Wavelet Network*, *International Journal of Robotics and Automation (IJRA)*, Vol. 3, No. 3, hal 191-200, (2014).
- [4] Ardisasmita, M.S., *Pengembangan Robot Mobil Otonom Menggunakan Sistem Kendali Fuzzy dan Jaringan Syaraf Tiruan*, *Risalah Lokakarya Komputasi dalam Sains dan Teknologi Nuklir XIV*, hal 157-170, (2003).
- [5] Wang, H., Chen, C., dan Huang, Z., *Ultrasonic Sensor Based Fuzzy-Neural*

*Control Algorithm of Obstacle Avoidance for Mobile Robot*, Springer-Verlag Berlin Heidelberg, Part 1, hal 824-833, (2007).

- [6] Aribowo, F.W., Fatchur, A., Setiawan, I., *Robot Mobile Penjejak Arah Cahaya Dengan Kendali Logika Fuzzy*, Transmisi, Jurnal Teknik Elektro, Jilid 10, No. 3, hal 144-150, (2008).
- [7] Adriansyah, A., Gunardi, Y., Badaruddin, dan Ihsanto, E., *Goal-Seeking Behavior-Based Mobile Robot Using Particle Swarm Fuzzy Controller*, Telkomnika, Vol. 13, No. 2, hal 528-538, (2015).
- [8] Pitowarno E., *Robotika : Desain, Kontrol, dan Kecerdasan Buatan*, Andi Offset, (2006).
- [9] Ping, J., Xiao-fang, H., dan Ai-dong, G., *Mobile Robot gas Source Localization Based on Behavior Strategies*, Proceeding of the 33<sup>rd</sup> Chinese Control Conference, hal 8304-8308, (2014).
- [10] Sugiarto, I., Untung, L.L., dan Rahman, M.I., *Implementation of Fuzzy Logic in FPGA for Maze Tracking of a Mobile Robot Based on Ultrasonic Distance Measurement*, Jurnal Teknik Elektro, Vol. 8, No.2, hal 96-102, (2008).
- [11] Fahmizal dan Kuo, C., *Development of a Fuzzy Logic Wall Following Controller for Steering Mobile Robots*, Proceedings of International Conference on Fuzzy Theory and Its Application, hal 7-12, (2013).
- [12] Farooq, U., Khalid, A., Amar, M., Habiba, A., Shafique, S., dan Noor, R., *Design and Low Cost Implementation of a Fuzzy Logic Controller for Wall Following Behavior of a Mobile Robot*, International Conference on Signal Processing Systems, vol. 2, hal 740-746, (2010).
- [13] Rendyansyah, Rivai, M., dan Purwanto, D., *Implementasi kendali Logika Fuzzy Untuk Mengontrol Pergerakan Olfactory Mobile Robot Dalam Mengikuti Dinding Obyek*, Prosiding Seminar Nasional Matematika dan Pendidikan Matematika, hal 384-395, (2015).
- [14] Darwison dan Wahyudi, R., *Kontrol Kecepatan Robot Hexapod Pemadam Api Menggunakan Metoda Logika Fuzzy*, Jurnal Nasional Teknik Elektro, Vol. 4, No. 2, hal 227-234, (2015).

### **Biodata Penulis**

**Rendyansyah**, lahir di Inderalaya, Sumatera Selatan, Indonesia. Pendidikan SD, SMP dan SMA di Kabupaten Ogan Ilir, Indonesia. Pada tahun 2007 melanjutkan kuliah di jurusan Sistem Komputer Universitas Sriwijaya dan memperoleh gelar sarjana komputer pada tahun 2011. Pernah bekerja sebagai tenaga laboratorium kendali dan robotika pada fakultas ilmu komputer UNSRI tahun 2011 sampai 2013. Tahun 2013 melanjutkan pendidikan magister teknik elektronika ITS Surabaya, dan selesai pada tahun 2015. Kemudian diterima sebagai tenaga pengajar S1 sistem komputer UNSRI.

**Kemahyanto Exaudi**, lahir di Lampung, Bangka Belitung, Indonesia. Pendidikan SD, SMP dan SMA di Bangka Belitung, Indonesia. Pada tahun 2008 melanjutkan kuliah di jurusan Sistem Komputer Universitas Sriwijaya dan memperoleh gelar sarjana komputer pada tahun 2011. Pernah bekerja sebagai tenaga laboratorium kendali dan robotika pada fakultas ilmu komputer UNSRI tahun 2011 sampai 2013. Tahun 2013 melanjutkan pendidikan magister teknik elektronika ITS Surabaya, dan selesai pada tahun 2015. Kemudian diterima sebagai tenaga pengajar S1 sistem komputer UNSRI.